Bolt Beranek and Newman Inc.

Report No. 4620

Research on Narrowband Communications

Quarterly Progress Report No. 2
18 November 1980—17 February 1981

Prepared for:
Defense Advanced Research Projects Agency

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

BBN

1. REPORT NUMBER: Report No. 4620 ✓

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): (6) RESEARCH ON NARROWBAND COMMUNICATIONS.

5. TYPE OF REPORT & PERIOD COVERED: Quarterly Prog. Rep. No. 2, 18 Nov. 1980 - 17 Feb. 1981

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): (10) John Makhoul, Salim Roukos, Michael Krasner, Richard Schwartz, John Sorensen

8. CONTRACT OR GRANT NUMBER(s): (15) F19628-80-C-0165

9. PERFORMING ORGANIZATION NAME AND ADDRESS: Bolt Beranek and Newman Inc., 10 Moulton St., Cambridge, MA 02230

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: ARPA Order-3515

11. CONTROLLING OFFICE NAME AND ADDRESS: (14) BBN-4620

12. REPORT DATE: March 1981

13. NUMBER OF PAGES: 30 (12) 35

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Deputy for Electronic Technology (RADC/EEV), Hanscom AFB, MA 01731, Mr. Anton Segota, Contract Monitor

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report): Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce, for sale to the general public.

(9) Quarterly progress rept. no. 2, 18 Nov 80 - 17 Feb 81,

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES: This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 3515, AMD. 4.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number): Speech compression, linear prediction, clustering, spectral template, vocoder, hierarchical clustering, unsupervised learning, diphone, phonetic vocoder, phoneme recognition, multiple speaker phonetic synthesis.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

This document reports on work toward a very low rate vocoder. We model speech as a Markov Chain of spectral templates for the unsupervised learning approach to very low rate vocoding. This quarter investigated some variations in the spectral clustering algorithms. We also decided to use the speech of many speakers for the clustering and sequential modeling. We also began work on synthesizing the speech of many speakers from a diphone data base recoreded from a single speaker. In phonetic

cont

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

060100

recognition, we compared two methods for "training" the diphone network, and concluded that the distance metric needs to be improved.

Report No. 4620

RESEARCH ON NARROWBAND COMMUNICATIONS

Quarterly Progress Report No. 2
18 November - 17 February 1981

Prepared by:

Bolt Beranek and Newman Inc.
10 Moulton Street
Cambridge, Massachusetts 02238

Prepared for:

Defense Advanced Research Projects

TABLE OF CONTENTS

1. SUMMARY

In this Quarterly Progress Report, we present our work performed during the period November 18 to February 17, 1981.

1.1 Introduction

The work in the past quarter was in the areas of an unsupervised learning approach to very-low-rate (VLR) vocoding, phonetic synthesis for many speakers, and phonetic recognition.

1.2 Unsupervised Learning

Our first unsupervised learning approach to VLR vocoding will be to generate a network of possible spectral sequences. As mentioned in QPR1, we will first generate a Markov chain of spectral template classifications.

During this quarter, we investigated some more detailed aspects of the behavior of spectral clustering algorithms. In particular:

- We determined that we could cluster the speech of several male voices using only 0.75 more bits than for the single speaker for the same spectral error.
- We investigated a technique called "cascaded" spectral

clustering that requires much less memory than full binary clustering.

- We performed some experiments to determine how much of the gain in clustering could be explained by principal components analysis and optimal scalar quantization of the resulting eigenvectors.

In considering the Markov chain, we determined that we need to use the multispeaker data base in order to have sufficient data for estimating the transition probabilities of a Markov chain.

Some of the effort this quarter was spent in moving all of our clustering and unsupervised learning programs to the VAX.

1.3 Multiple Speaker Synthesis

In the last part of this quarter, we started work on a number of techniques to alter the output speech of the phonetic synthesizer to sound like other speakers. A detailed report of these techniques and the results of our multi-speaker synthesis effort will be continued in the next QPR.

1.4 Phonetic Recognition

Our phonetic recognition work this quarter focused on comparing two different methods of training for the diphone

network. We determined that, at the present time, the approach of simply adding new training speech as alternate diphone templates in the network produced better results than the automatic training method. We will be modifying our automatic training procedure to be somewhat more similar to this simpler procedure.

We performed some experiments to determine the limitations in the phonetic recognition procedure. Our main conclusion was that the spectral distance metric was too simplistic, and that we need to try some measures that correlate more closely with perceptual judgements of phonetic similarity.

2. CLUSTER ANALYSIS

2.1 Introduction

During this quarter, we continued work on clustering speech spectra. A major effort was spent, during December 1980, to move the clustering programs to the VAX. This transfer is complete. Both programs and data files were shipped as ASCII files by Mag tape. The programs have been modified to work correctly under VMS.

The work on clustering consisted of several independent topics that we will discuss in this chapter. An important conclusion for this quarter is that we can use the multispeaker database to estimate a Markov model for speech spectra. We have started work to determine this Markov model.

2.2 Single Speaker Data

We repeated the results of the program for binary clustering for a single speaker using a large training set (~13000 frames). As shown in Table 1, entropy coding will have a negligible effect, that is, a savings of only 0.19 bits for 6 bits.

For every cluster, we also determined the percentage of

n	1	2	3	4	5	6	7	8	9	10
$\overline{e^2}$ dB	19.8	18.3	17.2	15.9	15.0	14.2	13.5	12.7	11.9	11.0
H	0.825	1.99	2.95	3.92	4.86	5.81	6.76	7.70	8.64	9.59

TABLE 1. Mean square error and entropy for a single speaker.

voiced and unvoiced frames that belong to the cluster. This allows us to infer from the spectrum alone whether a given frame is voiced or unvoiced. We present the results as a plot of the function p(x), which is the percentage of clusters that have x voiced frames, where x is the ratio of voiced frames to total frames in the cluster. If p (95%) = 30%, then 30% of the clusters have 95% of their frames voiced.

Figure 1 shows the function p(.) for 64 clusters. Note that 90% of the clusters have 85% of their frames either voiced or unvoiced. Hence, if we use the spectra to determine voicing, we will have an error rate of 15% on the voicing decision for 90% of the clusters. This error rate is large; however, the reason may be that the data have bad voicing information. From listening experiments, we found that both voicing and pitch had to be hand corrected for satisfactory intelligibility. We estimate the error to be around 15%. Further, even if we have to transmit a voicing decision for 10 to 20% of the templates, that would require a small bit rate. In the future,we will investigate the need to have any voicing information transmitted for our low bit rate vocoder. To determine the lowest bit rate required for speech spectra, we performed the following listening experiment.

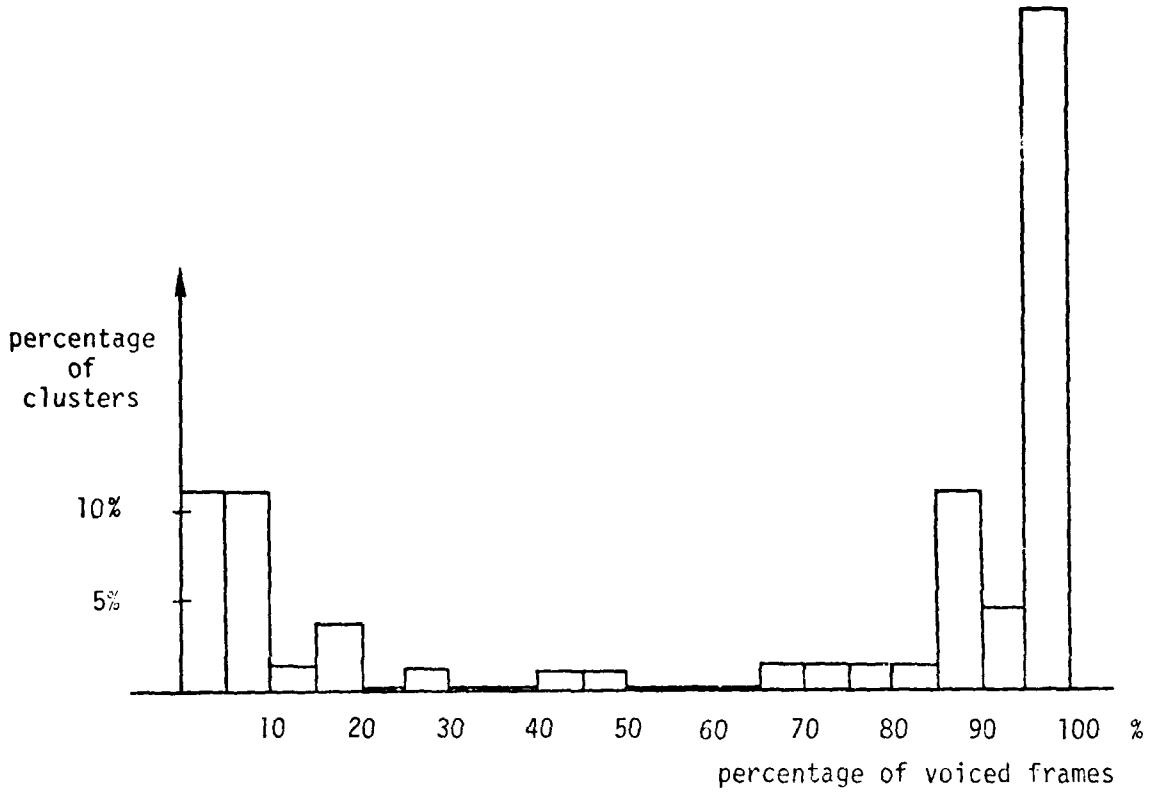

FIG. 1. Distribution of clusters determined by voicing.

2.3 Listening Test

To evaluate the intelligibility of a vocoder based on a 6 bit spectral representation, we performed the following informal experiment with 4 experienced listeners. Pitch and gain were unquantized in a LPC vocoder, but the spectrum was quantized to 64 templates (6 bits). We also used the variable-frame-rate algorithm with a rate of 34 frames/sec. The mean square quantization error of the LAR vectors was 27.9. We presented a set of six sentences from the Harvard list to the subjects. The sentences were played several times. The word error rate was 11%. This test suggests that the intelligibility, in context, may be acceptable for this spectral representation.

We are currently investigating how to reduce the bit rate of the above vocoder. One approach is to model speech as a Markov chain.

2.4 Clustering of Multispeaker Data

We do not currently have enough data for one speaker to determine a Markov Chain for speech. In addition, we would like to know whether it will be possible to generate a very-low rate system that can be used for many speakers. Therefore, we decided

to try using some of our data taken from several speakers. The multispeaker data we used consisted of 30 sec of speech for each of 20 speakers (~60,000 frames). The speakers read either a passage or a list of short phrases. We expect the data to be about one-third silence. We also have 13000 frames from a single speaker reading a subset of the Harvard sentences. All speakers were male. The speech was low-passed at 4900 Hz, high-passed at 60 Hz and sampled at 10 kHz with no preemphasis. The speech was analyzed using a 20 msec Hamming window with 10 msec overlap. Each frame (20 msec) was represented by 14 log area ratios (LAR), gain and pitch. The available data were 75,000 frames.

The multispeaker data were used to determine how many additional bits are required to represent speech spectra from male speakers for the same clustering error obtained for a single speaker. To compare our results to the single speaker case, we used the same training set size (~15000 frames). We used a 10 sec segment from each of 15 speakers. The speech segment was 1 of 6 different speech segments: 2 possible texts and 3 possible 10-sec segments from a total of 30 sec available. Hence, not only were the speakers different, but the speech was also different.

The results of binary clustering using a Euclidean distance metric are shown in Fig. 2. We have plotted the mean square

error for the multispeaker data and for a single speaker data. There was a relatively constant increase of 0.75 bits for the multispeaker case in order to get the same error rate as for the single speaker. This small increase indicates that, for male speakers, the clustering vocoder can handle multiple speakers with minimal increase in bit rates. We cannot evaluate the perceived quality yet because we do not have the synthesis programs and D/A capability on the VAX (where the experiment was performed). Also it is possible for the higher order statistics to change across speakers and produce a Markov model with a high entropy. However, since the increase in spectral error is small, we have decided that we will use the multispeaker data base for the initial Markov chain modeling.

2.5 Cascaded Clustering

There are two disadvantages to clustering: (1) the training database size grows exponentially with the number of bits, and (2) the required storage for the templates also grows exponentially with the number of bits. The limited training data set size is a severe problem. The required training set size is

$$M = \alpha L 2^n$$

where L is the dimensionality of the data, and n is the number of bits. A good rule of thumb in pattern recognition, requires α

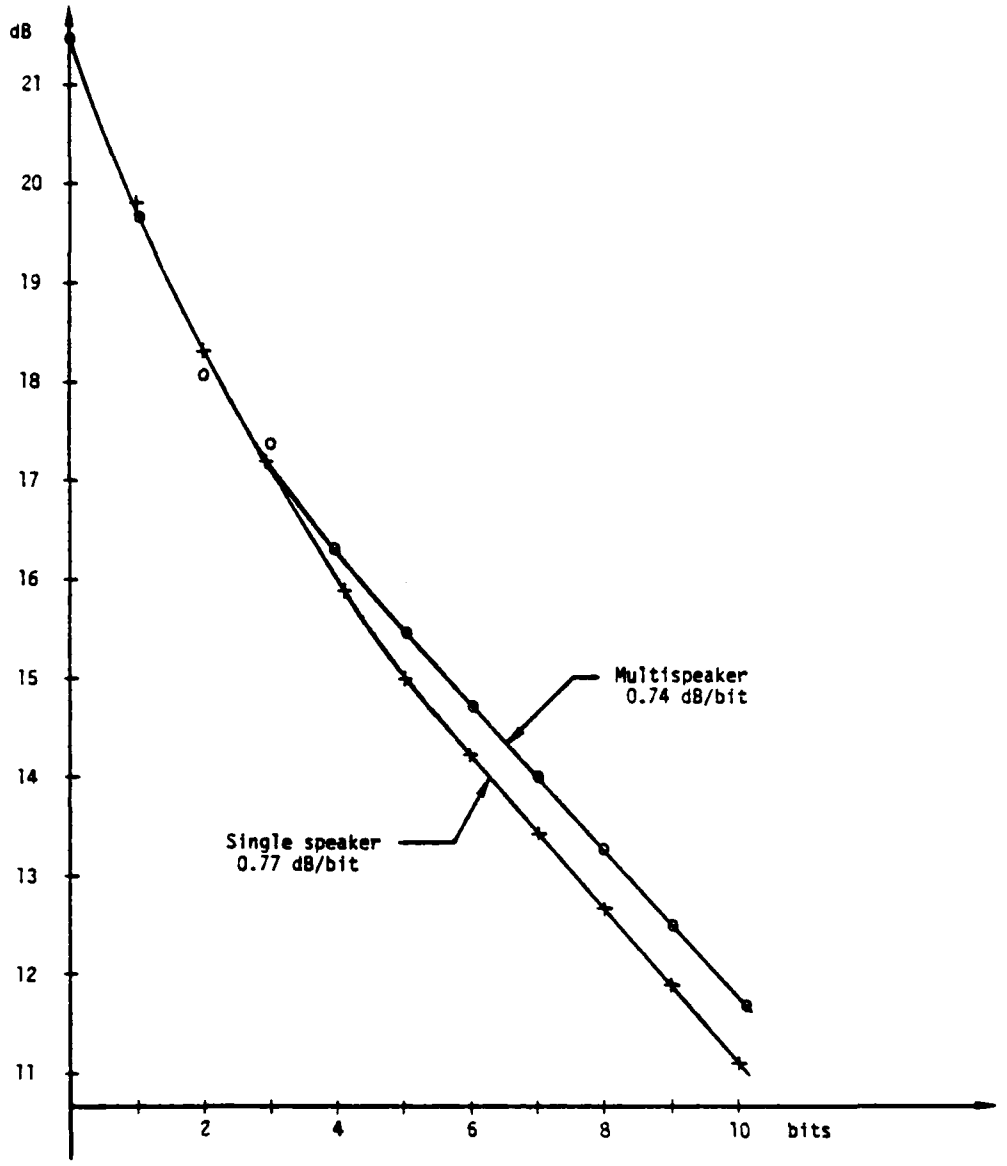

FIG. 2. Binary clustering error for single speaker data and multispeaker data.

(α=5 to 10) spectra per dimension for every cluster to get reliable statistical estimates. For example, 1 hr. of speech data is sufficient for no more 11 to 12 bits of clustering.

To avoid both difficulties, we attempted to use "cascaded clustering". First, an initial n-bit clustering, which we call the n-bit stage, is performed. Next, the deviations of the data from their templates are computed. A second m-bit stage clustering is performed on the data defined by the deviations. By clustering the deviations from all the clusters together, we are implicitly assuming that all clusters have the same deviations (or shape); hence, they are equivalent, and we can infer the statistics of each cluster by the average over all clusters. To partially satisfy this assumption, we represent the deviations of each cluster along the principal components of the corresponding cluster. Then we group all deviations. This corresponds to rotating the clusters so that their principal components align before superimposing them. To test the preceding ideas, we performed the following experiments.

1. Cascade 1-bit Stage

We divided the database into 2 clusters (1 bit). For each cluster, we computed the deviations from the template, combined all the deviations in a single set, and divided again into two

clusters. We repeated this process until we reached the required number of bits.

2. Cascade 1-bit Stage With Rotation

We performed the first experiment with the following additional step. Before combining the deviations for all clusters, we computed the eigenvectors (principal components) of the covariance matrix of each individual cluster. Then we represented the deviations of each cluster along the corresponding principal components. Finally, we combined the deviations and performed another 1-bit stage of clustering. We repeated this process for the required number of bits.

3. Cascade 4-bit With Rotation

In this case, instead of dividing the data into 2 clusters (1 bit), we performed 4 bits of clustering (16 clusters). Then we computed the deviations and rotated them.

Figure 3 shows the mean square error for the three preceding experiments. The important result is that all 3 methods reach the asymptote of 6 dB/(bit/dimension) or 0.43 dB/bit very quickly as compared to binary clustering. One may expect the poor performance of cascaded clustering; since by combining the deviations, we reduced the dependency structure of the data. Also, we can see that the cluster rotations yield a two bit gain by making use of some of the dependencies.

Since cascaded clustering is inefficient, it should only be used when clustering reaches the asymptote of 6 dB/(bit/dimension). To predict the performance of clustering and in particular the asymptotic behavior of clustering, we analyzed a bit allocation model.

2.6 Bit Allocation and Clustering

In the previous quarterly report, we found that the clustering error decreased at the rate of 0.86 dB/bit. For a 14 dimensional problem, the rate of 6 dB/(bit/dimension) for the uniform cube corresponds to 0.43 dB. We inferred that our speech data have an intrinsic dimensionality of 7. By considering a problem of bit allocation, we have a better understanding of our data and the "effective" dimensionality.

2.6.1 Bit allocation

Let the mean square error of each component of a random vector be

$$\overline{e_i^2}(B) = \sigma_i^2 h(B) \qquad (1)$$

where σ_i^2 is the variance of i-th component, B is the number of

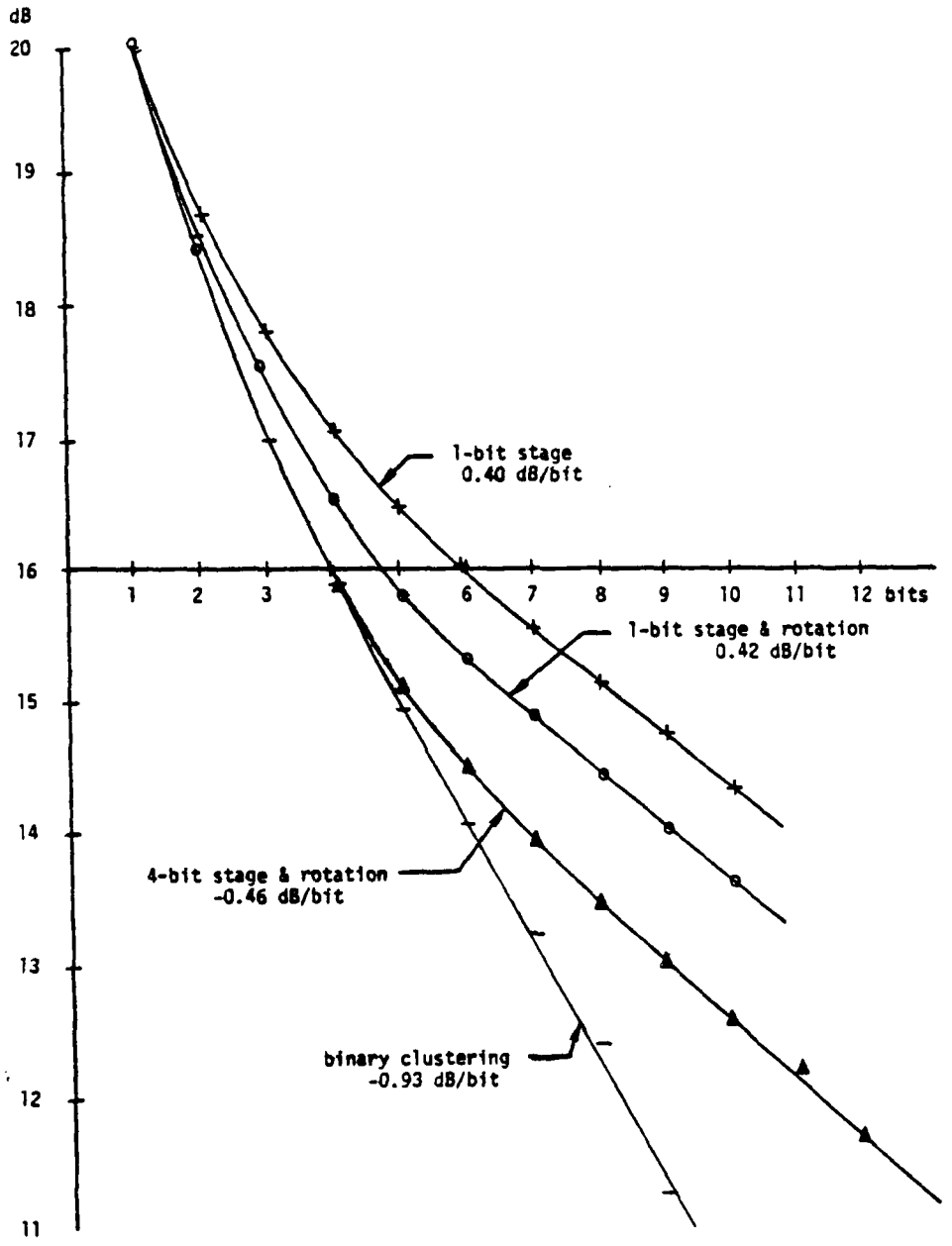

FIG. 3. Cascaded clustering compared to binary clustering.

bits used to quantize it, and h(.) is continuously differentiable and monotone decreasing. The bit allocation problem is to allocate a total of B bits to L random variables such that the total error:

$$\overline{e^2}(B) = \sum_{i=1}^{L} \sigma_i^2 h(B_i) \tag{2}$$

is minimized subject to the constraint:

$$\sum_{i=1}^{L} B_i = B \text{ and } B_i \geq 0 \tag{3}$$

where B_i bits are allocated to the i-th random variable. The solution for gaussian random variables was given in [1]. We discuss one aspect of the solution. To get a minimum, we must have, for all j such that $B_j > 0$,

$$\sigma_j^2 h'(B_j) = \text{constant} = \lambda \tag{4}$$

where h' is the derivative of h. Let g be the inverse function of h' (assumed to exist). The bits are allocated according to

$$B_j = g\left(\frac{\lambda}{\sigma_j^2}\right) \tag{5}$$

and the mean square error for the j-th component is

$$\overline{e_j^2}(B_j) = \epsilon_j^2 h\left(g\left(\frac{\lambda}{\sigma_j^2}\right)\right). \tag{6}$$

Let $h(g(x))=\alpha x$, then

$$\overline{e_j^2}(B_j) = \alpha\lambda = \text{constant if } B_j>0. \tag{7}$$

Hence, the mean square error, for all components with nonzero bit allocation, is the same. Those components with $B_j=0$ have a smaller error which is equal to their variance. The condition $h(g(x))=\alpha x$ is satisfied by $h(B)=2^{-2B}$, the 6 dB/bit rule. For this case, we have the following results:

1. Let B^* be the required number of bits for the total error to be $L\,\epsilon_{min}^2$ (i.e., all components have the same quantization error that is equal to the smallest variance). Then

$$B^* = \tfrac{1}{2}L(\lg_2 g - \lg_2 \epsilon_{min}^2) \tag{8}$$

where g is the geometric mean of the variances.

2. Let B_s be the required number of bits for total error to be equal to $L\epsilon_{min}^2$ when quantizing without bit allocation, then

$$B_s = \tfrac{1}{2}L(\lg_2 a - \lg_2 \epsilon_{min}^2) \tag{9}$$

where a is the arithmetic mean of the variances.

3. The gain due to bit allocation is

$$B_s - B^* = \frac{L}{2}\lg_2\left(\frac{a}{g}\right). \tag{10}$$

At B^*, the rate of decreases of the total error will be 6 dB/bit (bit is average bit per dimension). This is the same slope as scalar quantization without bit allocation. For our speech data, represented along the eigenvectors, we get

$$\begin{aligned} B^* &= \frac{14}{2}(1.84 + 0.57) = 16.92 \text{ bits} \\ B_s &= 27.40 \text{ bits} \\ B_s - B^* &= 10.48 \text{ bits} \end{aligned} \tag{11}$$

Thus, we expect, according to the bit allocation model, that clustering will have a slope of 0.43 dB/bit instead of 0.86 dB/bit at around 17 bits.

2.6.2 Results on Bit Allocation

The interpretation of clustering as a bit allocation scheme followed by scalar quantization is based on two assumptions.

1. Scalar quantization can achieve 6 dB/bit error decrease. This depends both on the quantizer and on the statistics of the parameters.
2. The structure (dependency) of the speech data can be explained by the covariance matrix of the LAR's.

To verify assumption 2, we did the following experiment. We

computed the principal components of the covariance matrix of our speech data (multispeaker, 5000 frames, 14 LAR). The geometric mean of the variances along the eigenvectors is smaller. Therefore, bit allocation on the eigenvectors is more effective [1]. We performed an n-bit clustering, using p principal components. The value of p was determined by bit allocation, using a total of n bits. Table 2 and Fig. 4 show the total mean square error and compare it to the usual binary clustering. The two curves are fairly close, indicating that the correlation between the log area ratios explains most of the dependency structure. We also show the curve obtained without performing the eigenvector analysis; it has slightly larger error. This increase indicates that the correlation between log area ratios can be used to decrease the quantization error.

A more interesting result is obtained by predicting the quantization error from the eigenvalues under the assumption of bit allocation and a 6 dB/bit quantizer. This curve is also shown in Fig. 4. We can see that the fit of this parametric result is surprisingly good. We assumed a continuous bit allocation. However, the assumption of a 6 dB/bit quantizer may prove to be optimistic. We still need to implement the optimal scalar quantizer with bit allocation and compare the performance to clustering. But, we already can see that bit allocation

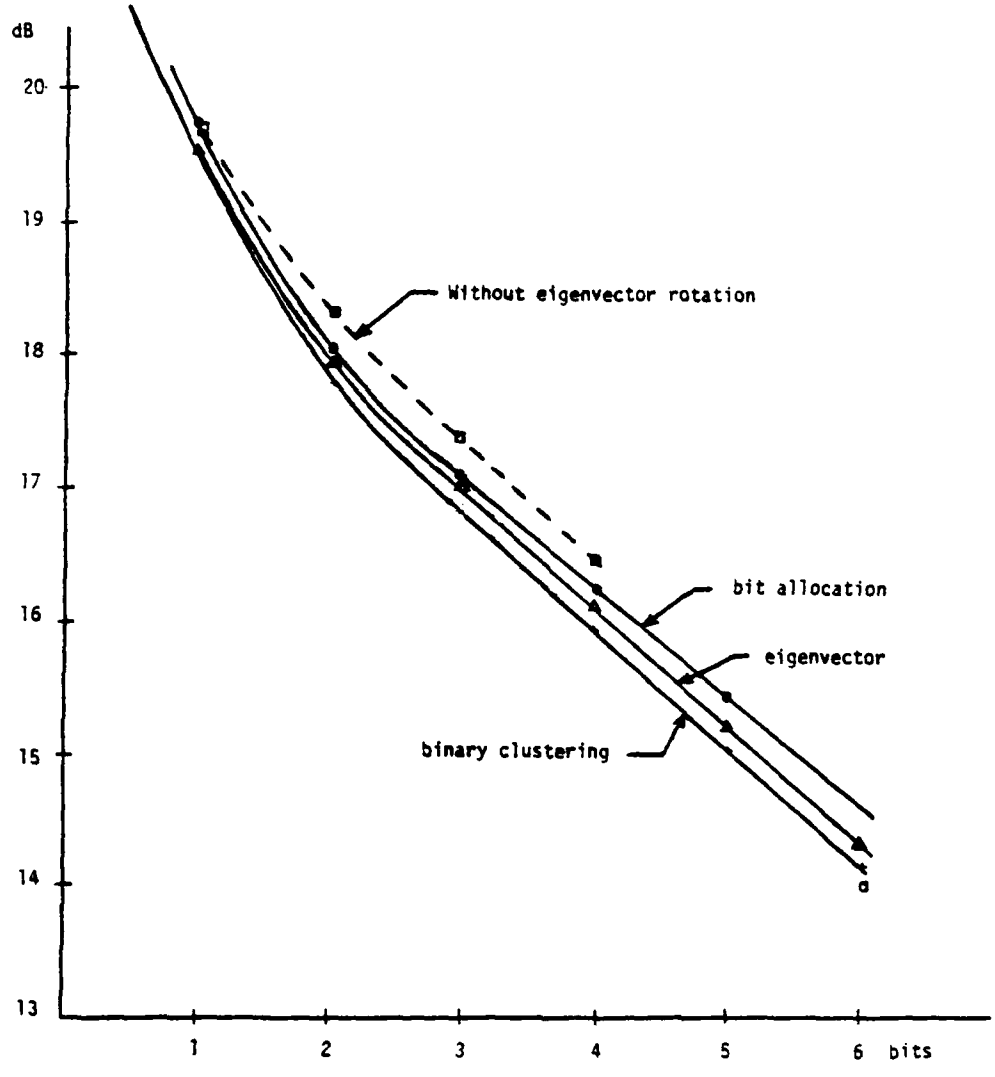

FIG. 4. A comparison of bit allocation to clustering. The dotted line shows the error of clustering using bit allocation on the log area ratios.

n		0	1	2	3	4	5	6
bit allocation	P		2	2	3	4	6	7
	$\overline{e^2}$ dB	21.50	19.79	18.05	17.12	16.25	15.43	14.03
Clustering on eigenvectors	P		2	2	3	4	6	7
	$\overline{e^2}$ dB	21.50	19.54	17.96	17.0	16.17	15.2	14.34
Binary clustering	$\overline{e^2}$ dB	21.50	19.53	17.89	16.81	15.91	15.02	16.17
Clustering without eigenvectors	P		2	2	3	4		
	$\overline{e^2}$ dB	21.50	19.73	18.33	17.4	16.43		

TABLE 2. The mean square error of binary clustering and bit allocation. The value of p indicates how many eigenvectors were used in the clustering.

explains most of the gain of clustering (note for n=6 bits, we quantize only 7 eigenvectors, from Table 2). This is mostly due to the unequal variances as compared to other statistical dependencies.

In the next section, we introduce a model for speech that is a Markov chain. We will use this model to reduce the bit rate of the clustering vocoder discussed in Section 2.3.

2.7 Markov Chain Model

The output of the VFR algorithm is a sequence of templates and durations of transitions from one template to another. During a transition, the log area ratios are linearly interpolated. We want to model the sequence of templates as a Markov chain. For the moment, we will not model the durations of the transitions.

The purpose of the Markov model is to reduce the bit rate with no loss in quality or intelligibility. The reduction is based on the fact that, given the present state (template), not all states (templates) can occur next. Hence, the conditional entropy satisfies

$$h(x_n | x_{n-1}=j) \leq H(x_n) \qquad (12)$$

where $H(x_n)$ is the unconditioned entropy of the state at time n. The entropy of the Markov chain is the expected value of h:

$$H_m = E_{x_n} h(x_n|x_{n-1}) \leq H(x_n). \quad (13)$$

Thus, the entropy of the Markov chain is smaller than (or equal to) the zero memory entropy of the source. To determine the reduction in bit rate, we need to estimate the transition probabilities p_{ij} from state i to state j. For 64 templates, we have 4096 possible transitions. Hence, to estimate the transition probabilities we need at least $40x10^3$ spectra, (an average of 10 occurrences for each transition). However, since this is the output of a VFR algorithm, (with an average transmission rate of 34 frames/sec) we need $120x10^3$ frames of speech. We do not have these data yet for a single speaker. However, if we use the multispeaker data, we have $75x10^3$ frames. We also know that the above 6-bit vocoder has about the same mean square error on the multispeaker data as for a single speaker. Hence, we will use the multispeaker data to estimate the Markov model. We do not expect the Markov model to be sufficient to reduce the bit rate to 100 bps; however, the resulting network, which represents the Markov Chain, would be a good starting point for merging paths to get the required bit rate. We have several approaches that we plan to develop in the near future. We are currently developing the Markov model.

3. MULTI-SPEAKER SYNTHESIS

This quarter we began working on being able to synthesize speech from our VLR vocoder (phonetic synthesizer) that will sound more like the speaker who is talking than like the speaker whose speech was used for the diphone templates. The transformation used the average vocal tract length (VTL) and long-term average spectrum taken from an arbitrary one minute speech sample for the new speaker. The preliminary results are encouraging.

4. PHONETIC RECOGNITION

During this quarter, we performed experiments to evaluate the automatic training capability implemented last quarter [2]. The automatic training capability allows the researcher to input to the matcher a sentence that has been phonetically transcribed. The input transcription includes both phonetic labels and time markers. The matcher then finds the best alignment (and corresponding score) of the input utterance against the network under the constraint of the transcription. Once completed, the matcher uses the input utterance to "train" the network. Those portions of the input utterance that are similar (closer than a preset threshold distance using the metric) to some path in the network are used during the training procedure by updating the statistics of that closest path in the network to include the input utterance parameters. The statistics of the network path that are modified include the the means and variances of the LARs and the PDFs of the frame durations. The remaining portions of the input utterance, those which are not very similar to any existing path in the network, are used to add alternate branches to the network. The parameters of those portions of the input utterance are used to create new branches of the network. By this procedure of updating network statistics and augmenting the network with new branches, we ensure that the network can match

the training data within the specified error threshold. We hypothesize that when the network is well trained with natural speech utterances (rather than the initial data base composed of diphone templates extracted from nonsense utterances), that the network will have sufficient paths and accurate statistics to model arbitrary input utterances well.

To evaluate the above training method, we "trained" the network on several sentences, and then tested the updated (trained) network using several other sentences. During the training procedure, it was found that for some portions of utterances, the matcher could not align the input speech to the network with the exact time constraints given by the transcription. This situation is possible with the current system because the matcher will not align more than two node spectra with one input frame spectrum. Thus, if a phone in the input training speech was of much shorter duration than the corresponding original diphone templates such that the input had less than half as many frames as the network had nodes, the alignment procedure would fail. To circumvent this problem, we allowed the matcher (at user request) to add those diphones from the input speech directly into the network as alternate diphone branches. This addition of diphone branches is implemented in the same manner as "augment mode" of the network compiler

described below. After the addition of diphone branches, the input utterance is realigned and the training procedure restarted. Although this simple algorithm modification fixes the problem and allows the training procedure to work properly, two basic modifications to the algorithm would eliminate the cause of the problem and are suggested as future changes:

1. The matcher should allow more than two network nodes to be aligned to a single input frame. This alignment would incur an appropriate score penalty.

2. The automatic training procedure would probably be more efficient (and effective) processing one diphone at a time. This would eliminate the process of finding the optimal alignment for an entire utterance.

Another method of updating the network that we implemented relies on augmenting the network with additional diphone branches. In this procedure, we use the transcription of the training data, together with the network compiler program, to create new alternate diphone templates. There are three differences of this augmentation algorithm with the previously described training method:

1. The entire diphone is added as an alternate path, rather than accepting some preset error as in the training algorithm.

2. In augment mode, the compiler assumes that the diphone boundary is at the middle of the labeled phone (which is a good heuristic) rather than letting the program assign the diphone boundary where it chooses.

3. In augment mode, there are no a priori probabilities assigned to paths. This differs from the training mode where several paths may be "averaged" together. These a priori probabilities, however, are not currently used by the matcher.

A comparison of the results using the training algorithm and augmenting algorithm show 8% better phoneme recognition with the augmentation method. From these results, we conclude that:

1. The error threshold described for automatic training should be lowered.
2. The alignment procedure should also constrain the alignment of the diphone boundaries.

With these changes, and with the additional use of the a priori probabilities, the automatic training method should work better.

To assess the likely performance of the phonetic recognizer, we carefully examined the results of some experiments performed in April of 1980. In these experiments, we recorded two instances of a paragraph. The first instance was transcribed and used for training in the "augment" mode to create an updated network. The matcher then was used to test this network on the second instance of the same paragraph. In this test, 79% of the phonemes were recognized correctly. It should be noted that this is an overestimate of the eventual (with training) performance of the current system (current algorithm with the currently-used metric), since the diphones used in training in this experiment

were always in the same context as they were in the test sentences. Therefore, the diphones in the network were more similar to the test diphones than generally would be the case after training.

Careful examination of the errors showed that the spectra (of the training data and test data) were indeed significantly different when measured using a mean-square spectral error (or euclidean distance between LAR vectors) as we use. In most cases, however, the spectral sequences were "phonetically similar". That is, the peaks in the spectrum were at roughly the same frequencies (though different amplitudes) and the formant trajectories and spectral differences over time were quite similar. This observation reinforces our previous belief that an error measure that correlates more closely with human judgement of phonetic differences is needed to improve recognition performance. The metric would include a comparison of the changes over time of the template and input, in addition to the steady-state frame-by-frame differences used now. By studying the errors of the present algorithm in the context of the present spectral metric and human speech perception, we have identified several attributes for an improved metric. We will be testing and evaluating the resultant "phonetic metrics" in the coming months.

REFERENCES

1. Segall, A., "Bit Allocation and Encoding for Vector Sources," IEEE Trans. Inform. Theory, Vol. IT-22, March 1976, pp. 162-169.

2. Makhoul, J., Roukos, S. and Schwartz, R., "Research on Narrowband Communications," Bolt Beranek and Newman Inc., Quarterly Progress Report No. 4557, 18 Aug. to 17 Nov. 1980.